

# HELP!!1! THE DATABASE IS SLOW!!

ANDRÁS VÁCZI



# /me

- with Postgres since 2007
- mostly a DBA
- main mission is trying to be helpful



# Who is Cybertec?



# What is this *not*?

- no one-size-fits-all solution!



# Who is this for?

- you are a DBA (and actually have contact to developers)



# Who is this for?

- you are a DBA (and actually have contact to developers)
- you are a developer (and have contact to a/the DBA)



# Who is this for?

- you are a DBA (and actually have contact to developers)
- you are a developer (and have contact to a/the DBA)
- you are a DevOps engineer (and can only ask yourself)



# Who is this for?

- you are a DBA (and actually have contact to developers)
- you are a developer (and have contact to a/the DBA)
- you are a DevOps engineer (and can only ask yourself)
- you are a support engineer







# Boooom!!!

- things were working fine



# Boooom!!!

- things were working fine
- now they aren't



# Boooom!!!

- things were working fine
- now they aren't
- possible 'reports':



# Boooom!!!

- things were working fine
- now they aren't
- possible 'reports':
  - 'The application doesn't work'



# Boooom!!!

- things were working fine
- now they aren't
- possible 'reports':
  - 'The application doesn't work'
  - 'Customer service can't even...'



# Boooooom!!!

- things were working fine
- now they aren't
- possible 'reports':
  - 'The application doesn't work'
  - 'Customer service can't even...'
  - 'The database is slow!!'



# How to be more helpful

- be as specific as possible





# How to be more helpful

- be as specific as possible
  - impact



# How to be more helpful

- be as specific as possible
  - impact
    - ▶ customer facing?



# How to be more helpful

- be as specific as possible
  - impact
    - ▶ customer facing?
    - ▶ report cannot be created for CEO?



# How to be more helpful

- be as specific as possible
  - impact
    - ▶ customer facing?
    - ▶ report cannot be created for CEO?
    - ▶ dev system somewhat slow?



# How to be more helpful

- be as specific as possible
  - impact
    - ▶ customer facing?
    - ▶ report cannot be created for CEO?
    - ▶ dev system somewhat slow?
  - description that others can understand as well



# How to be more helpful

- be as specific as possible
  - impact
    - ▶ customer facing?
    - ▶ report cannot be created for CEO?
    - ▶ dev system somewhat slow?
  - description that others can understand as well
  - what you have checked already



# How to be more helpful

- be as specific as possible
  - impact
    - ▶ customer facing?
    - ▶ report cannot be created for CEO?
    - ▶ dev system somewhat slow?
  - description that others can understand as well
  - what you have checked already
- BUT: you don't have to know everything



# So, something *is* slow

- let's fix it!





DB is slow



do something



DB is fast



# But wait, is it really the DB?

- observability FTW!!!



# But wait, is it really the DB?

- observability FTW!!!
  - metrics and monitoring



# But wait, is it really the DB?

- observability FTW!!!
  - metrics and monitoring
  - alerting



# But wait, is it really the DB?

- observability FTW!!!
  - metrics and monitoring
  - alerting
  - logging



# But wait, is it really the DB?

- observability FTW!!!
  - metrics and monitoring
  - alerting
  - logging
  - tracing



# Metrics

General / Health-check

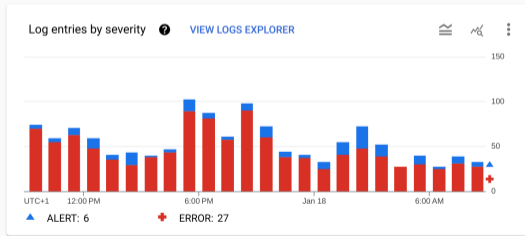
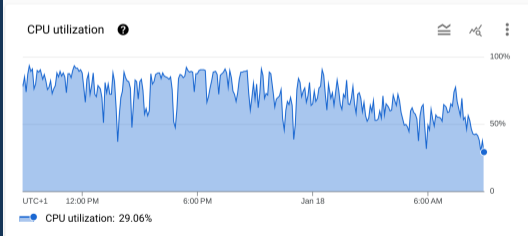
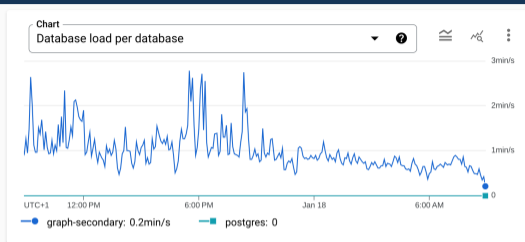
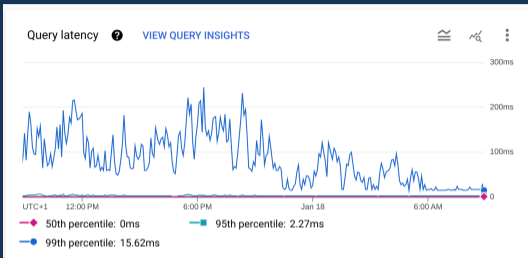
Last 30 minutes

dbname: test | Max. age for 'online' metrics: 10m

Instance state	Instance uptime	PG version num.	Longest query runtime
PRIMARY	10 months	150005	31 seconds
Active connections	Max. connections	Blocked sessions	Shared Buffers hit pct.
5	100	0	100.0%
TX rollback pct. (avg.)	TPS (avg.)	QPS (avg.)	'Idle in TX' count
0%	1.7	8.0	0
DB size (last)	DB size change (diff.)	DATADIR disk space left	Query runtime (avg.)
25.8 MiB	72.0 KiB	30.4 GiB	107.3 ms
Config change events	Table changes	WAL archiving status	WAL folder size
0	0	OK	48.0 MiB
Invalid / duplicate indexes	Autovacuum issues	Checkpoints requested	Approx. table bloat
0	0	0	1.6 MiB
WAL per second (avg.)	Temp. bytes per second (avg.)	Longest AUTOVACUUM duration	Seq. scans on >100 MB tables per minute (avg.)
2.2 KiB	96.4 KiB	N/A	N/A
INSERT-s per minute (avg.)	UPDATE-s per minute (avg.)	DELETE-s per minute (avg.)	Backup duration
60	188	0	N/A

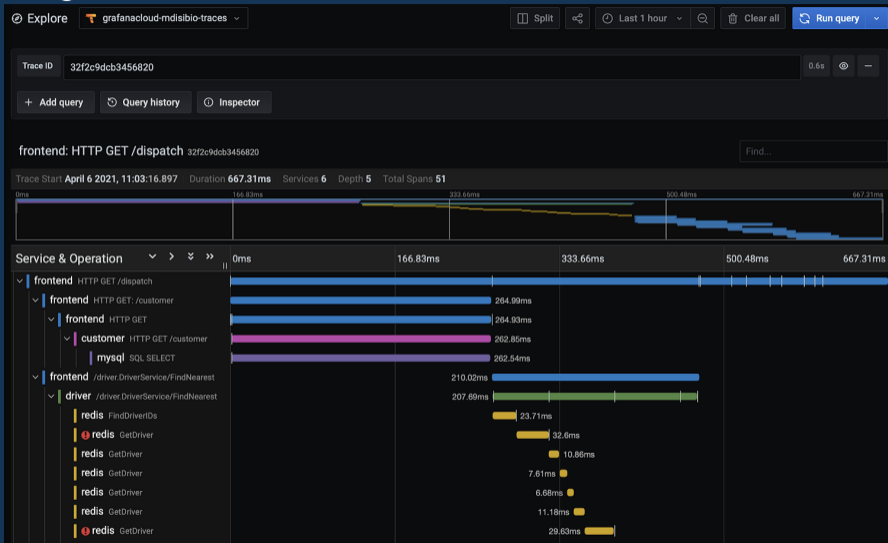


# Metrics





# Tracing



# It's not the DB

- can be the app



# It's not the DB

- can be the app
- can be the network



# It's not the DB

- can be the app
- can be the network
- is the fan in your laptop blowing like crazy?



# It's not the DB

```
fetch_plate_number(id):  
    conn.execute("SELECT plate_number FROM car WHERE id = :id")  
    plate_number = conn.result()  
    sleep(1)  
    return plate_number
```



# It's not the DB (?)

- waiting for connections



# It's not the DB (?)

- waiting for connections
  - slowness



# It's not the DB (?)

- waiting for connections
  - slowness
  - timeouts





# It's not the DB (?)

- waiting for connections
  - slowness
  - timeouts
  - configuration issues?



# It's not the DB (?)

- waiting for connections
  - slowness
  - timeouts
  - configuration issues?
    - ▶ `max_connections`



# It's not the DB (?)

- waiting for connections
  - slowness
  - timeouts
  - configuration issues?
    - ▶ `max_connections`
    - ▶ pooler settings

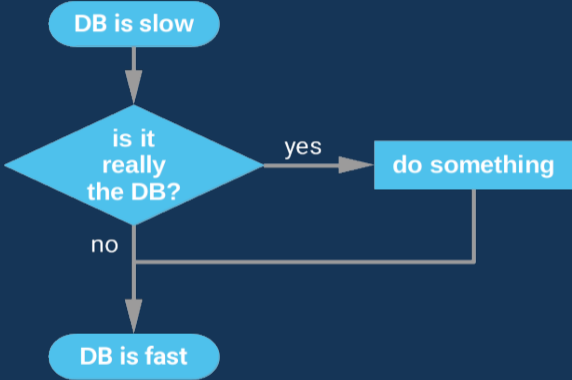


# It's not the DB (?)

- waiting for connections
  - slowness
  - timeouts
  - configuration issues?
    - ▶ `max_connections`
    - ▶ pooler settings
    - ▶ no pooling

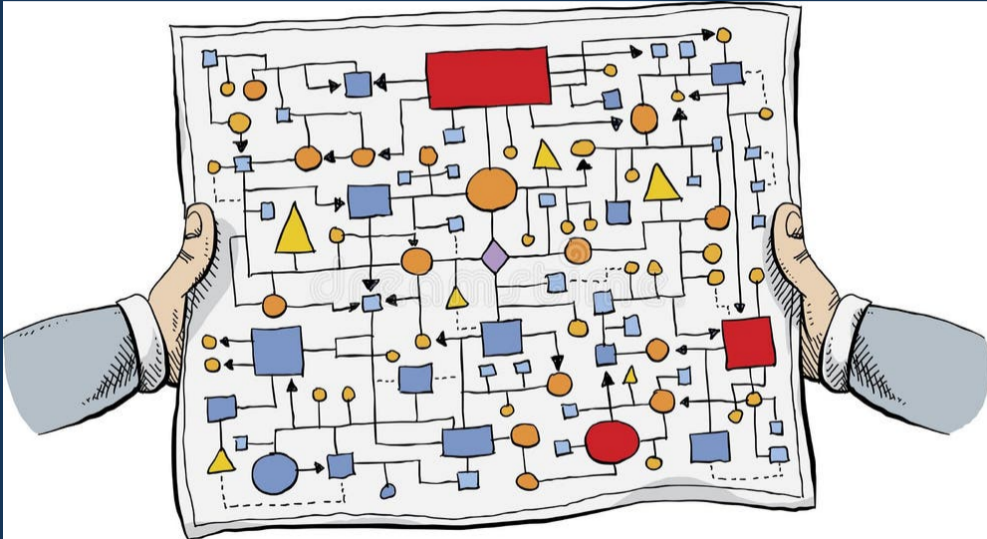


# Recap: where are we now?



# Well, it really seems to be the DB

- fix it!



# How bad is it?

- single query affected (localized symptoms)



# How bad is it?

- single query affected (localized symptoms)
- everything is slow



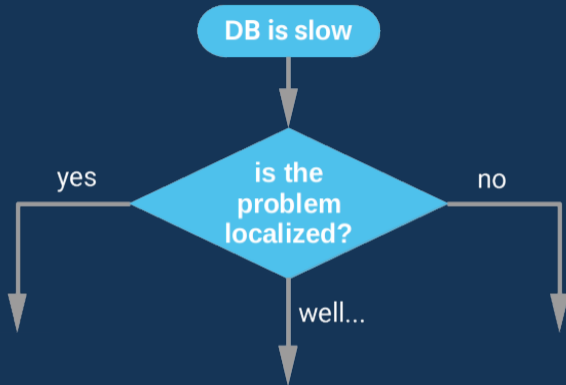




# How bad is it?

- single query affected (localized symptoms)
- everything is slow
- some stuff is running fine, some other isn't





# Single query - but which one?

- the developer might know



# Single query - but which one?

- the developer might know
- ORM might know



# Single query - but which one?

- the developer might know
- ORM might know
- application logs might at least give a hint



# Single query - but which one?

- the developer might know
- ORM might know
- application logs might at least give a hint
- Postgres logs might know



# Single query - but which one?

- the developer might know
- ORM might know
- application logs might at least give a hint
- Postgres logs might know
  - `log_min_duration_statement`





## Side track: log visibility

Logs might contain passwords, query parameters, table and column names, etc.

Some of this might be protected by policy, law, or even common sense. Visibility should be restricted the same way access to the DB itself is restricted.



# We caught it, now what?

- let's fix it!



DB is slow



do something



DB is fast



... come on...

- you can start with ANALYZE



... come on...

- you can start with ANALYZE
- with luck you'll see it just looking at the query



... come on...

- you can start with ANALYZE
- with luck you'll see it just looking at the query
  - nicely formatted query



... come on...

- you can start with ANALYZE
- with luck you'll see it just looking at the query
  - nicely formatted query
  - full table definition



... come on...

- you can start with ANALYZE
- with luck you'll see it just looking at the query
  - nicely formatted query
  - full table definition
- EXPLAIN





Table "public.film"

Column	Type	Collation	Nullable	Default
film_id	integer		not null	nextval('film_film_id_seq'::regclass)
title	character varying(255)		not null	

Indexes:

"film\_pkey" PRIMARY KEY, btree (film\_id)

"idx\_title" btree (title)

Foreign-key constraints:

"film\_language\_id\_fkey" FOREIGN KEY (language\_id) REFERENCES language(language\_id) ON UPDATE CASCADE ON DELETE RESTRICT

Referenced by:

TABLE "film\_actor" CONSTRAINT "film\_actor\_film\_id\_fkey" FOREIGN KEY (film\_id) REFERENCES film(film\_id) ON UPDATE CASCADE ON DELETE RESTRICT

Triggers:

last\_updated BEFORE UPDATE ON film FOR EACH ROW EXECUTE FUNCTION last\_updated()



# EXPLAIN me in more detail

- a slow query is often doing a lot of unnecessary work



# EXPLAIN me in more detail

```
Delete on request (actual time=1750842.493..1750842.493 rows=0 loops=1)
  -> Seq Scan on request (actual time=0.816..0.83189.199 rows=19836585 loops=1)
        Filter: ((request_type_id <> 9000) AND (date_creation < now() - '2 days'::interval))
        Rows Removed by Filter: 19836585
```

- Probably an index is missing



# EXPLAIN me in more detail

```
Sort (actual time=3033.815..3033.817 rows=4 loops=1)
```

```
  Buffers: shared hit=36 read=6434
```

```
  I/O Timings: read=2887.963
```

```
  -> Limit
```

```
    -> Sort
```

```
      -> Index Scan myindex on mytable (actual time=432.562..3033.546 rows=4 loops=1)
```

```
        Buffers: shared hit=24 read=6434
```

```
        I/O Timings: read=2887.963
```

- there's an index, but not the right one



# EXPLAIN me in more detail

```
Sort (actual time=3033.815..3033.817 rows=4 loops=1)
  Buffers: shared hit=36 read=6434
  I/O Timings: read=2887.963
  -> Limit
    -> Sort
      -> Index Scan myindex on mytable (actual time=432.562..3033.546 rows=4 loops=1)
        Buffers: shared hit=24 read=6434
        I/O Timings: read=2887.963
```

- there's an index, but not the right one
- table bloat



# EXPLAIN me in more detail

- conditions not SARGable



# EXPLAIN me in more detail

- conditions not SARGable
  - rewrite query?



# EXPLAIN me in more detail

- conditions not SARGable
  - rewrite query?
- wrong index used





# EXPLAIN me in more detail

- conditions not SARGable
  - rewrite query?
- wrong index used
  - raise statistics target



# EXPLAIN me in more detail

- conditions not SARGable
  - rewrite query?
- wrong index used
  - raise statistics target
  - CREATE STATISTICS

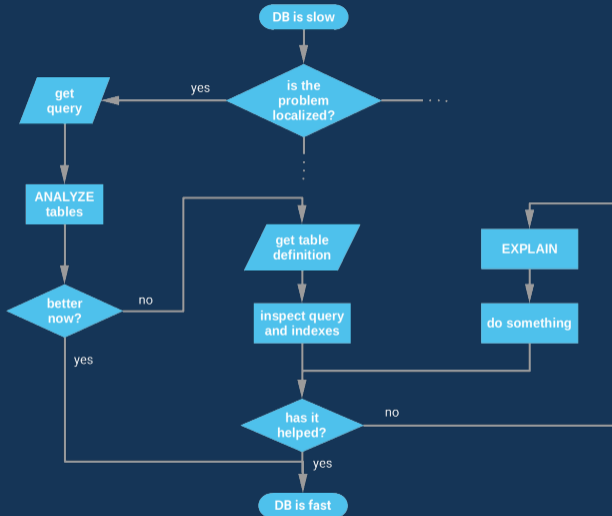


# EXPLAIN me in more detail

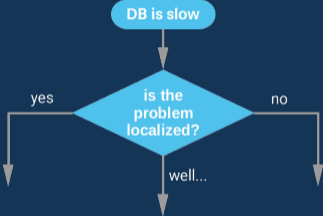
- conditions not SARGable
  - rewrite query?
- wrong index used
  - raise statistics target
  - CREATE STATISTICS
  - ...



# Recap so far



# Everything is slow



# What we might see

- long waits



# What we might see

- long waits
  - `log_lock_waits`



# What we might see

- long waits
  - `log_lock_waits`
- no free connections left (see It's not the DB (?))





# What we might see

- long waits
  - `log_lock_waits`
- no free connections left (see It's not the DB (?))
  - `superuser_reserved_connections` to the rescue



# What we might see

- long waits
  - `log_lock_waits`
- no free connections left (see It's not the DB (?))
  - `superuser_reserved_connections` to the rescue
- too many *active* connections



# What we might see

- long waits
  - `log_lock_waits`
- no free connections left (see It's not the DB (?))
  - `superuser_reserved_connections` to the rescue
- too many *active* connections
- there may be less prominent queries still running fine

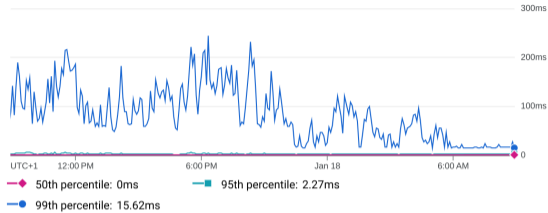


# What we might see

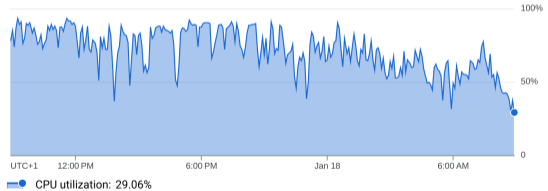
- long waits
  - `log_lock_waits`
- no free connections left (see It's not the DB (?))
  - `superuser_reserved_connections` to the rescue
- too many *active* connections
- there may be less prominent queries still running fine
  - p50/p95/p99 latency - maybe even p99.9



Query latency ? [VIEW QUERY INSIGHTS](#)



CPU utilization ?



# I am waiting at the counter...

- locks held for too long



# I am waiting at the counter...

- locks held for too long
  - idle in trasaction



# I am waiting at the counter...

- locks held for too long
  - idle in trasaction
    - ▶ `idle_in_transaction_session_timeout`





# I am waiting at the counter...

- locks held for too long
  - idle in transaction
    - ▶ `idle_in_transaction_session_timeout`
    - ▶ use Postgres logs and `pg_stat_activity` to identify the source



# I am waiting at the counter...

- locks held for too long
  - idle in transaction
    - ▶ `idle_in_transaction_session_timeout`
    - ▶ use Postgres logs and `pg_stat_activity` to identify the source
  - long running transactions



# I am waiting at the counter...

- locks held for too long
  - idle in transaction
    - ▶ `idle_in_transaction_session_timeout`
    - ▶ use Postgres logs and `pg_stat_activity` to identify the source
  - long running transactions
    - ▶ application logic?



# I am waiting at the counter...

- locks held for too long
  - idle in transaction
    - ▶ `idle_in_transaction_session_timeout`
    - ▶ use Postgres logs and `pg_stat_activity` to identify the source
  - long running transactions
    - ▶ application logic?
    - ▶ analytics done on primary?



# I am waiting at the counter...

- locks held for too long
  - idle in transaction
    - ▶ `idle_in_transaction_session_timeout`
    - ▶ use Postgres logs and `pg_stat_activity` to identify the source
  - long running transactions
    - ▶ application logic?
    - ▶ analytics done on primary?
    - ▶ big updates done during peak hours?

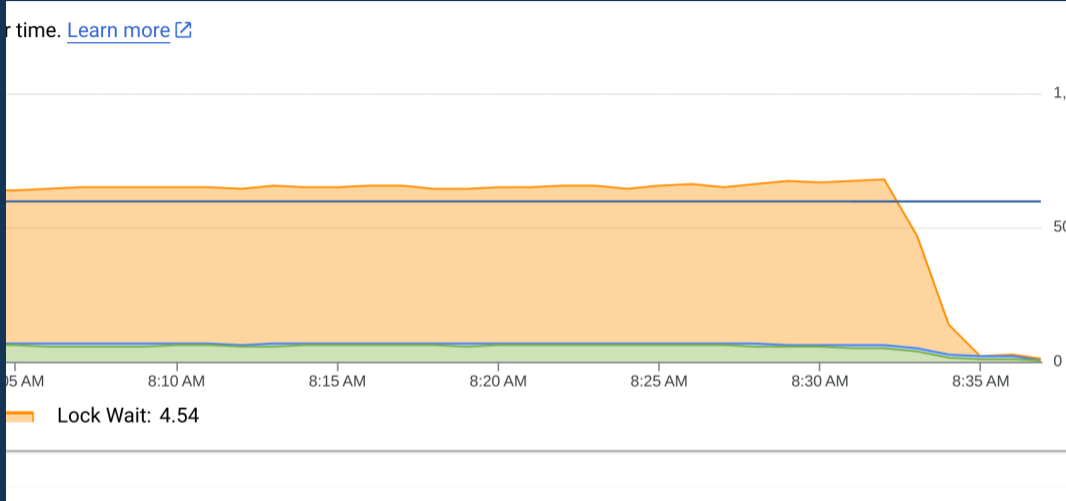


# I am waiting at the counter...

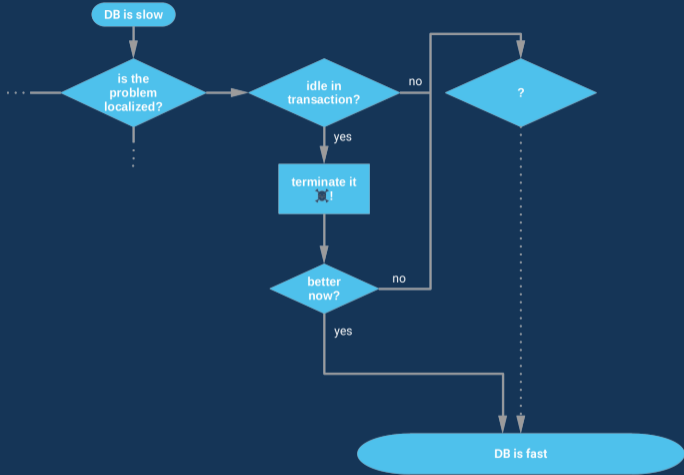
- locks held for too long
  - idle in transaction
    - ▶ `idle_in_transaction_session_timeout`
    - ▶ use Postgres logs and `pg_stat_activity` to identify the source
  - long running transactions
    - ▶ application logic?
    - ▶ analytics done on primary?
    - ▶ big updates done during peak hours?
    - ▶ these make a DB unhealthy



# Idle in transaction



# Idle in transaction





... for the man to pour the coffee

- locks held for too long



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX
    - ▶ REFRESH MATERIALIZED VIEW



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX
    - ▶ REFRESH MATERIALIZED VIEW
    - ▶ ALTER TABLE



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX
    - ▶ REFRESH MATERIALIZED VIEW
    - ▶ ALTER TABLE
    - ▶ ...



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX
    - ▶ REFRESH MATERIALIZED VIEW
    - ▶ ALTER TABLE
    - ▶ ...
  - writes on a central table



# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX
    - ▶ REFRESH MATERIALIZED VIEW
    - ▶ ALTER TABLE
    - ▶ ...
  - writes on a central table
    - ▶ may lead to bloat



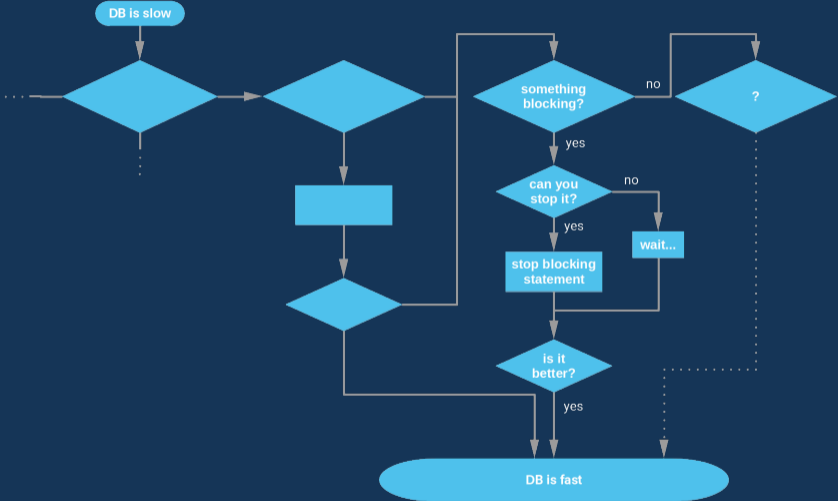


# ... for the man to pour the coffee

- locks held for too long
  - straight out blocking
    - ▶ CREATE INDEX
    - ▶ REFRESH MATERIALIZED VIEW
    - ▶ ALTER TABLE
    - ▶ ...
  - writes on a central table
    - ▶ may lead to bloat
    - ▶ will slow down a lot of things



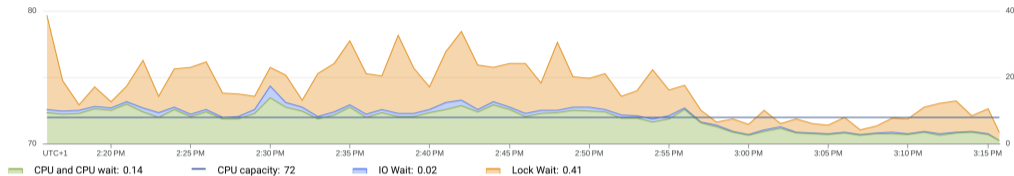
# Blocking



# Bloat on a central table

## Database load - all queries

A measure of the work (in CPU-seconds) that all executed queries in your selected database perform over time. [Learn more](#)



## Top queries and tags

An overview of the queries and tags that cause the most database load within the data and time range currently selected. For a closer look at a specific query's details, select one. [Learn more](#)

QUERIES

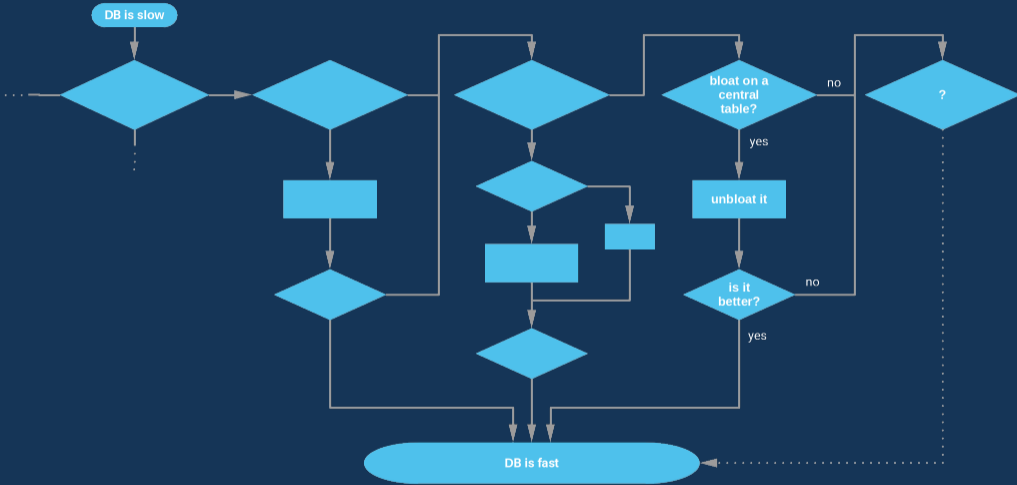
TAGS

Filter Filter queries

Query	Database	Load by total time <span>▼</span> <span>↓</span>	Avg execution time (ms)	Times called	Avg rows returned
UPDATE		<div style="width: 100%; height: 10px; background-color: orange;"></div>	16.65	1,445,849	1
SELECT		<div style="width: 20%; height: 10px; background-color: orange;"></div>	2.62	1,430,874	0
SELECT		<div style="width: 10%; height: 10px; background-color: orange;"></div>	2.86	786,058	1



# Bloat on a central table



# Too many (active) connections

- queries are slow



# Too many (active) connections

- queries are slow
  - handle them, maybe there's a common theme (missing index, etc.)



# Too many (active) connections

- queries are slow
  - handle them, maybe there's a common theme (missing index, etc.)
- Christmas is coming



# Too many (active) connections

- queries are slow
  - handle them, maybe there's a common theme (missing index, etc.)
- Christmas is coming
  - undersized HW





# Too many (active) connections

- queries are slow
  - handle them, maybe there's a common theme (missing index, etc.)
- Christmas is coming
  - undersized HW
  - prepare for peak times



# Sudden drop of query performance

- new piece of workload



# Sudden drop of query performance

- new piece of workload
- shift in data distribution



# Sudden drop of query performance

- new piece of workload
- shift in data distribution
- cloud setup, IOPS limit reached



# Sudden drop of query performance

- new piece of workload
- shift in data distribution
- cloud setup, IOPS limit reached
  - quick workaround: increase storage size a bit



# Sudden drop of query performance

- new piece of workload
- shift in data distribution
- cloud setup, IOPS limit reached
  - quick workaround: increase storage size a bit
    - ▶ terms and conditions apply



# Sudden drop of query performance

- new piece of workload
- shift in data distribution
- cloud setup, IOPS limit reached
  - quick workaround: increase storage size a bit
    - ▶ terms and conditions apply
- server restart, cold cache



# Sudden drop of query performance

- new piece of workload
- shift in data distribution
- cloud setup, IOPS limit reached
  - quick workaround: increase storage size a bit
    - ▶ terms and conditions apply
- server restart, cold cache
- storage level issues



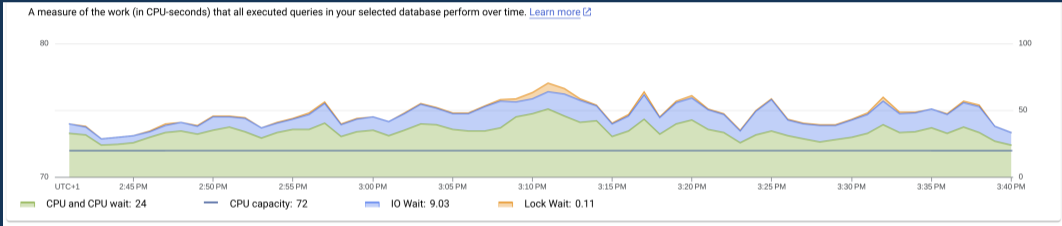


# Sudden drop of query performance

- new piece of workload
- shift in data distribution
- cloud setup, IOPS limit reached
  - quick workaround: increase storage size a bit
    - ▶ terms and conditions apply
- server restart, cold cache
- storage level issues
  - rare footage



# I/O wait





# Or a gradual worsening

- data volume is growing



# Or a gradual worsening

- data volume is growing
- doing too much I/O?



# Or a gradual worsening

- data volume is growing
- doing too much I/O?
  - too little RAM / small `shared_buffers` or `work_mem`



# Or a gradual worsening

- data volume is growing
- doing too much I/O?
  - too little RAM / small `shared_buffers` or `work_mem`
  - bad cache hit ratio



# Or a gradual worsening

- data volume is growing
- doing too much I/O?
  - too little RAM / small `shared_buffers` or `work_mem`
  - bad cache hit ratio
  - reading unnecessary data





# Or a gradual worsening

- data volume is growing
- doing too much I/O?
  - too little RAM / small `shared_buffers` or `work_mem`
  - bad cache hit ratio
  - reading unnecessary data
    - ▶ `SELECT *`



# CPU load plateau

- everything looks good in isolation and at calmer times?



# CPU load plateau

- everything looks good in isolation and at calmer times?
  - maybe time to increase CPU core count



# CPU utilization ?



CPU utilization: 93.02%



Thank you!



# Feedback

- feedback helps us improve our talks
- feedback helps to make the conference even better
- login with your postgresql.org account (i.e. how you registered for the conference)



# Contact

With critique, hints, questions, and everything – please drop me a message at [andras.vaczi@cybertec.at](mailto:andras.vaczi@cybertec.at)

